



ECDR-GC814-FV-A VxWorks Driver User's Manual

Revision 2.0

ECHOTEK CORPORATION

555 Sparkman Drive
Suite 400
Huntsville, AL 35816
Phone: (256) 721-1911
Fax: (256) 721-9266
E-Mail: Sales@echotek.com
Web Site: <http://www.echotek.com>

Copyright statement:

All rights reserved.

Disclaimer:

The information in this document has been carefully checked and is believed to be entirely reliable. However, no responsibility is assumed for inaccuracies. Echotek Corporation makes no representations or warranties with respect to the contents hereof and specifically disclaims any implied warranties of merchantability or fitness for any particular purpose. Furthermore, Echotek Corporation reserves the right to make changes to any product herein to improve reliability, function or design, without obligation of Echotek Corporation to notify any person of such revision or changes. Echotek Corporation does not assume any liability arising out of applications or use of any product or circuit described herein; neither does it convey any license under its patent rights nor the rights of others.

Revision History

Rev.	Date	Chapter/Section	Change/Addition
1.0	4-15-2004	All	Initial release
2.0	5-6-2004		

TABLE OF CONTENTS

Revision History.....	iii
CHAPTER 1 INTRODUCTION.....	1-1
Features.....	1-2
Before Getting Started.....	1-3
Technical Support.....	1-4
CHAPTER 2 COMPILING AND LINKING	2-1
Compiling and Linking the Driver Under UNIX.....	2-2
CHAPTER 3 INSTALLING THE DRIVER.....	3-1
Installing the ECDR-GC814-FV-A Driver.....	3-2
CHAPTER 4 FUNCTION CALLS.....	4-1
open.....	4-2
read.....	4-3
close	4-4
ioctl	4-5
CHAPTER 5 FLASH PROGRAMMING PROCEDURE	5-1
Running the Flash Programming Software.....	5-2

Chapter 1 Introduction

This chapter includes a brief introduction to Echotek Corporation's ECDR-GC814-FV-A vxWorks driver, a list of system requirements, and contact information necessary for requesting technical support.

Features

The ECDR-GC814-FV-A vxWorks Driver is designed for use with Echotek's ECDR-GC814-FV-A board under the vxWorks real-time operating system. This driver is supplied as source code; after installation under vxWorks, it supports the following Application Program Interface (API) calls:

- open –opens the ECDR-GC814-FV-A for reading.
- read – performs a read from the ECDR-GC814-FV-A.
- close – closes the file descriptor.
- ioctl – performs miscellaneous operations.

For complete descriptions of the function calls, refer to Chapter 4 Function Calls.

Before Getting Started

Before installing the ECDR-GC814-FV-A driver, make sure that your system meets the following requirements:

- The ECDR-GC814-FV-A board is installed in your system.
- If the ECDR-GC814-FV-A is configured for use with an external clock source, connect the clock input to the ECDR-GC814-FV-A. The input clock should have an amplitude range of between -2dBm and +4dBm.



Any clock source with amplitude outside this range can cause the board to malfunction and may cause possible damage to the board.

- The analog input connected to the ECDR-GC814-FV-A should have an amplitude of less than or equal to +5dBm.



Any input with amplitude exceeding +5dBm causes the Analog-to-Digital converter to saturate and can cause possible damage to the board.

Technical Support

If you need additional technical information or assistance, contact Echotek's support department:

E-Mail:	support@echotek.com
Telephone:	256.721.1911
Facsimile:	256.721.9266
Address:	Echotek Corporation 555 Sparkman Drive, Suite 400 Huntsville, Al 35816

Chapter 2 Compiling and Linking

This chapter provides you with information for compiling the ECDR-GC814-FV-A driver object files and linking the user code under the UNIX operating system.

Compiling and Linking the Driver Under UNIX

Follow the procedure below to compile the driver under the UNIX operating system.

1. Unzip **ECDR-GC814-FV-A_Drv_vXpX.tar.gz**, where **XpX** is the current revision of the driver.
2. Untar **ECDR-GC814-FV-A_Drv_vXpX.tar**, where **XpX** is the current revision of the driver.
3. To recompile the driver, run make commands at the command prompt in the following order:

```
make clean  
make  
make driver
```

4. Then link your code (ex: a.out) with the ECDR-GC814-FV-A driver to create a relocateable object named `final.out` using a Power PC linker. For example, key in the following to link the DRIVER object with a.out:

```
ldppc -t -o final.out -r a.out DRIVER
```

Chapter 3 Installing the Driver

This chapter provides steps for installing the ECCR-GC814-FV-A driver under vxWorks.

Installing the ECDR-GC814-FV-A Driver

The utility `ecdr814gcInstall("DeviceName", hardwareAddress)` must be run before any software that uses the driver, since it is used to detect the presence of the ECDR-GC814-FV-A board. After the board is found, the utility creates a device under vxWorks named `DeviceName` or `ECDR814GC`, if `DeviceName` is not provided.

Follow these instructions to install the ECDR-GC814-FV-A Driver.

1. Under vxWorks, load the final object file (the object file consisting of the driver and the user application):

```
ld < final.out
```

Or if you decide to load the driver object and the user application object separately, do the following:

```
ld < DRIVER
ld < UserObject.out
```

2. After loading the driver successfully, run the `ecdr814gcInstall()` installation routine. Note that `hardwareAddress` corresponds to the hardware address of the board.

```
ecdr814gcInstall("DeviceName", hardwareAddress)
```

3. After installing the ECDR-GC814-FV-A driver successfully, use the ECDR-GC814-FV-A driver function calls in *Chapter 4 Function Calls* to run the ECDR-GC814-FV-A board.

Chapter 4 Function Calls

This chapter describes the function calls supported by the ECDR-GC814-FV-A driver.

open

Opens an ECDR-GC814-FV-A device as specified by name, and if successful, returns the associated file descriptor. You specify the name of the ECDR-GC814-FV-A in the `ecdr814gcInstall` call. If the name of the ECDR-GC814-FV-A is unspecified, the name defaults to "ECDR814GC."

Synopsis

```
#include "ecdr814gcDrv.h"
```

```
int open(const char *name,  
         int flags,  
         int unused)
```

Return Values

Success = Returns the file descriptor associated with the ECDR-GC814-FV-A device.

Failure = `EC_ERROR`

Example

```
void DemoFunc(void)  
{  
    int Fd;  
    .  
    .  
    .  
    Fd = open("ECDR814GC", O_RDONLY, 0);  
    if(Fd == EC_ERROR)  
        printf("ERROR: ECDR-GC814-FV-A could not be opened\n");  
    .  
    .  
    .  
}
```

read

Reads from the ECDR-GC814-FV-A. The ECDR is polled for data available if poll is set to 1, otherwise, it waits on a data available interrupt. The number of samples requested is indicated by samples.

Synopsis

```
#include "ecdr814gcDrv.h"
```

```
int read( int fd,  
          char *poll,  
          size_t samples);
```

Return Values

Success = EC_OKAY

Failure = EC_ERROR

Example

```
int DemoFunc(int Fd, int *poll, int samples)  
{  
    int Status = 0;  
  
    Status = read(Fd, (char *) poll, samples);  
    if(Fd == EC_ERROR)  
        printf("ERROR: could not read from ECDR-GC814-FV-A \n");  
}
```

close

Frees the file descriptor and closes the opened ECDR-GC814-FV-A device.

Synopsis

```
#include "ecdr814gcDrv.h"
```

```
STATUS close(int Fd);
```

Return Values

Always returns EC_OKAY.

Example

```
void DemoFunc(int Fd)
{
    close(Fd);
}
```


ioctl

Performs miscellaneous access to and control of the ECDR-GC814-FV-A board. This function uses a function code to indicate what action to perform on the board. Refer to Table 4-1 for the available ioctl function codes and the argument(s) that are required for each.

When more than one argument is required, it should be placed in an integer array. For example, ECDR814_SET_BUFFER requires three arguments: channel, sub-channel, and data buffer.

This would be coded as follows:

```
int bufferInfo[3];
int fd;
int *dataBuffer;

bufferInfo[0] = 3; //ECDR channel 3
bufferInfo[1] = 2; //GC sub-channel 2
bufferInfo[3] = dataBuffer; //location to which data will be stored
ioctl(fd, ECDR814_SET_BUFFER, bufferInfo);
```

Synopsis

```
#include "ecdr814gcDrv.h"
```

```
int ioctl( int fd,
           int function,
           int arg)
```

Return Values

Success = EC_OKAY

Failure = EC_ERROR

Example

```
void DemoFunc(int Fd)
{
    /*-----*/
    /* -: Program all 6620s :- */
    /*-----*/

    Status = ioctl(hEC, ECDR814_PROGRAM_ALL_CHANNELS, 0);
    if(Fd == EC_ERROR)
        printf("ERROR: could not perform ECDR-GC814-FV-A ioctl\n");
}
```

Table 4-1 IOCTL Function Codes

IOCTL Commands	Argument	Command Descriptions
ECDR814_CHECK	None	Checks to see if data collection is complete.
ECDR814_CLEAR	None	Resets the data SRAM pointers.
ECDR814_DISABLE	None	Disables data collection.
ECDR814_GET_BYTE_COUNT	ByteCount address	Writes the number of bytes to be collected on each trigger to ByteCount.
ECDR814_GET_CHANNEL	ChannelNumber address	Writes the channel number to be used to ChannelNumber. Available ChannelNumber values are: CHAN_0 CHAN_1 CHAN_2 CHAN_3 CHAN_4 CHAN_5 CHAN_6 CHAN_7
ECDR814_GET_DATA_MODE	DataMode address	Writes the data mode to be used to DataMode. Available DataMode values are: COUNT_DATA PACKED_DATA RAW_AD_DATA UNPACKED_DATA
ECDR814_GET_NUM_BITS	Integer array containing: 1) ChannelNumber 2) NumberOfBits address	Writes the number of bits used by ChannelNumber to NumberOfBits. Available ChannelNumber values are: CHAN_0 CHAN_1 CHAN_2 CHAN_3 CHAN_4 CHAN_5 CHAN_6 CHAN_7
ECDR814_GET_NUM_SUB_CHANNELS	Integer array containing 1) ChannelNumber 2) SubChannelsUsed address	Writes the number of sub-channels used by ChannelNumber to SubChannelsUsed. Available ChannelNumber values are: CHAN_0 CHAN_1 CHAN_2 CHAN_3 CHAN_4 CHAN_5 CHAN_6 CHAN_7 Available SubChannelsUsed values are: ONE_CHANNEL TWO_CHANNELS FOUR_CHANNELS

Table 4-2 IOCTL Function Codes

IOCTL Commands	Argument	Command Descriptions
ECDR814_GET_TRANSFER_TYPE	TransferType address	Writes the transfer type to TransferType. Available TransferType values are: SINGLE_READ_TRANSFER VME_TRANSFER
ECDR814_MODIFY_WORD_COUNT	WordCount	Forces all DMAs to be of size WordCount.
ECDR814_PROGRAM_ALL_CHANNELS	None	Programs each channel based on the setup contained in the initialization file.
ECDR814_PROGRAM_CHANNEL	ChannelNumber	Programs ChannelNumber base on the setup contained in the initialization file. Available ChannelNumber values are: CHAN_0 CHAN_1 CHAN_2 CHAN_3 CHAN_4 CHAN_5 CHAN_6 CHAN_7
ECDR814_PROGRAM_FLASH	Integer array containing 1) FlashID 2) FlashFile	Programs FlashID with the information contained in FlashFile. Available FlashID values are: USE_FLASH_0 USE_FLASH_1 USE_FLASH_2 USE_FLASH_3
ECDR814_READ_INI_FILE	InitializationFile	Reads the initialization file specified by InitializationFile.
ECDR814_RELEASE	None	Releases all semaphores.
ECDR814_RESET	None	Resets the local bus and the packet counter.
ECDR814_SET_BUFFER	Integer array containing 1) ChannelNumber 2) SubChannel 3) Buffer	Associates Buffer to ChannelNumber and SubChannel for data storage. Available ChannelNumber values are: CHAN_0 CHAN_1 CHAN_2 CHAN_3 CHAN_4 CHAN_5 CHAN_6 CHAN_7 Available SubChannel values are: SUB_CHAN_0 SUB_CHAN_1 SUB_CHAN_2 SUB_CHAN_3
ECDR814_SET_CHANNEL	ChannelNumber	Writes ChannelNumber to be used. Available ChannelNumber values are: CHAN_0 CHAN_1 CHAN_2 CHAN_3 CHAN_4 CHAN_5 CHAN_6 CHAN_7

Table 4-3 IOCTL Function Codes

IOCTL Commands	Argument	Command Descriptions
ECDR814_SET_NUM_TRIGS	NumTrigs	Writes NumTrigs to be collected.
ECDR814_SHOW	None	Displays the contents of the ECDR registers
ECDR814_SHOW_PROGRAM_INFO	None	Displays the setup information obtained from an initialization file.
ECDR814_TRIG	None	Performs a software trigger if the ECDR is setup for software trigger mode.
ECDR814_TEST_FLASH	None	Performs a wrap test of the ECDR flash.
ECDR814_VERIFY_FLASH	Integer array containing 1) FlashID 2) FlashFile	Verifies FlashID against the information contained in FlashFile. Available FlashID values are: USE_FLASH_0 USE_FLASH_1 USE_FLASH_2 USE_FLASH_3

Chapter 5 Flash Programming Procedure

This chapter provides the procedure that you use to program the flash. The board will be delivered with the flash pre-programmed. **Do not** program the flash unless directed to do so by Echotek support.

Running the Flash Programming Software

Below are the procedures for running the flash programming software using a Motorola 5100 PowerPC computer with NFS enabled.

1. First load the driver and the demo code into vxWorks (launch from the ./etc directory).

```
ld <../bin/DRIVER
```

2. Next, run the `ecdr814gcInstall()` installation function with your hardware address.

```
ecdr814gcInstall("ECCR814GC", yourAddress)
```

3. Run the `ecdr814gcSetup` function with your program file name (including path and extension) along with the flash identifier.

Note: flashIdentifier 0 = channel pair 0 and 1
1 = channel pair 2 and 3
2 = channel pair 4 and 5
3 = channel pair 6 and 7

```
ecdr814gcSetup(yourProgramFileAndPath, flashIdentifier)
```

Several lines will scroll on the screen while programming is being done. When the function returns, programming is complete and power should be cycled on the board.